软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal ofSoftware, [doi: 10.13328/j.cnki.jos.000000]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

开源项目贡献者流失预测*

谢佳苗[1], 刘嘉琳[1]

[1](长沙理工大学 计算机与通信工程学院,湖南 长沙 410076)

[1](长沙理工大学 计算机与通信工程学院,湖南 长沙 410076)

摘 要: 在当今的软件开发领域，开源项目已成为技术创新和协作的核心力量。而这些项目的成功往往依赖于一个活跃的贡献者社区，他们的持续参与对于项目的活力和进步至关重要。然而，面临的一个重要问题是贡献者的流失，这不仅会减缓项目的发展速度，还可能削弱其创新潜力。本研究提出了基于多种机器学习和时间序列预测的方法，通过分析贡献者的行为数据、提交频率和社区互动等关键指标，来识别和预测贡献者流失的风险。我们的目标是为开源项目的维护者和相关组织提供一种工具，使他们能够及时识别并应对潜在的人才流失问题，从而保护和增强项目的长期稳定性。通过这种预防性的方法，我们希望能够为开源社区的健康发展提供支持，进而促进整个技术生态系统的进步。

关键词: 开发者流失预测;机器学习;时间序列预测

本文第 1 节介绍开源项目贡献者流失预测的研究背景和研究价值.第 2 节介绍本文的数据收集与预处理.第 3 节介绍数据挖掘工具及方法.第 4 节介绍健康度量、寿命预测的方法及其效果.第 5 节介绍实验结果.第 6 节介绍科学与实际意义

1 研究背景和研究价值

开源项目在现代软件开发中扮演着至关重要的角色，它们为开发者提供了丰富的代码库、工具和框架。随着云计算、大数据、人工智能等技术的快速发展，开源项目在推动技术创新和产业升级方面发挥着越来越重要的作用。

开源项目的成功与否很大程度上有赖于贡献者和公司的持续参与，比如Linux内核已经有超过 1400 家公司参与贡献。开发者流失可能导致项目进展缓慢，缺乏新功能或改进，以及安全漏洞的长时间悬而未决，公司的撤资或减少投入可能会影响项目长期可持续性，并且长期的贡献者流失可能导致项目陷入停滞状态，甚至面临被废弃的风险。因此，识别潜在的流失风险对开源项目的持续发展至关重要。

目前，如何吸引和留住贡献者成为开源项目管理的重要挑战之一。通过我们研究的模型能够对开发者贡献频率及相关因素进行有效分析，预测他们的流失风险，这样有助于项目管理团队提前采取措施，降低流失风险。比如采取增加激励、改善沟通、提供培训等措施，以留住贡献者。这同时有助于提高开源项目的长期可持续性和稳定性。

同时，根据预测结果，项目管理团队可以调整管理策略，如优化贡献者招募流程、提高贡献者参与度等，以提高项目的整体效率和质量。通过及时采取措施留住贡献者，可以保持开源项目的活跃度和竞争力，进而推动整个开源生态的繁荣和发展。

2 数据收集与预处理

本项目使用的数据集来源于GitHub的开源项目贡献记录，一是关于Linux内核开源软件项目的数据，二是围绕Rust基础开源软件项目的数据。为了让数据更加完整，我们通过GitHub API收集了从 2001 年 9 月 17 日至 2023 年 11 月 22 日期间Linux内核项目的所有提交记录(commits)和从 2010 年 6 月 23 日至 2021 年 12 月 6 日期间Rust项目的所有提交记录(commits)。记录中包含了以下字段：author_name、author_email、author_date、subject以及message。

2.1 数据加载与初步处理

我们首先利用从 GitHub API 中获取的一个包含提交记录的 JSON 文件中加载数据，并用 Python 的 pandas 库进行转换。转换后的数据包括提交者的提交时间、电子邮件地址、提交的代码数量等多个字段。为了确保数据的准确性和一致性，让后续更加方便的处理数据，我们首先删除了部分有缺失值的数据，并且将重复和无效的提交记录进行清理，然后将提交者的邮箱地址进行统一，将提交时间字段转换为标准的日期时间格式。最后将 pd.DataFrame 转换成 csv 文件进行保存。

2.2 特征工程

在开发者流失预测中，特征工程是构建有效预测模型的关键步骤，它涉及从原始数据中提取、构建和转换特征，以便机器学习模型能够更好地理解和预测目标变量-开发者是否会流失。我们通过创建时间序列提取了多种与时间相关的特征，并且我们还从原始数据中提取了开发者的相关行为活动特征。

·提取时间特征

我们从开发者的最后一次活动时间中提取了年、月、日、星期几和小时等时间特征。这些特征有助于我们识别开发者的活动模式，比如他们是否在特定的时间段内更为活跃。

·计算活动频率

通过按开发者的标识进行分组，并统计每个开发者的活动次数，我们得到了开发者的活动频率特征。这一特征能够反映开发者的参与热情和项目投入度。

·计算活跃周期

我们记录了每个开发者首次和最近一次活动的时间，并据此计算了他们在项目中的活跃周期（以天数计）。这有助于我们评估开发者的持续参与情况和可能的流失风险。

·分析开发者在社区中的行为

通过分析开发者在代码仓库、邮件列表、论坛等社交网络中的行为和互动模式。例如，开发者的社交网络中心性、群组参与度等，这些特征可以反映开发者在社区中的活跃度和影响力。

·分析代码质量

通过分析开发者提交的代码质量，如代码审查得分、单元测试通过率、代码覆盖率等，这些特征可以反映开发者的专业能力和对项目的贡献质量。

·分析贡献趋势

分析通过开发者的贡献趋势，如是否呈现上升趋势、下降趋势或周期性变化，这些特征有助于预测开发者的未来行为。

3 数据挖掘工具及方法

在进行开发者流失预测的项目中，我们依托了一系列先进的数据挖掘工具和技术来处理和分析数据：

·Python：作为核心编程语言，负责数据的采集、清洗、预处理以及模型的开发和测试。

·Pandas：用于数据的导入、处理和执行基础的统计分析，提供了便捷的数据结构和数据分析工具。

·NumPy：提供了高效的数值计算能力，特别适用于处理大型多维数组和矩阵，是执行科学计算的基础。

·scikit-learn：这是一个功能丰富的机器学习库，我们用它来实现和评估多种机器学习算法，包括用于预测开发者流失风险的模型。

在数据挖掘过程中，我们采用的方法包括：

·特征工程：通过深入分析开发者的行为数据（例如代码提交的频率、时间分布、代码审查参与度等），我们提取了多个关键特征，这些特征能够有效地预测开发者的流失风险。

·分类算法：我们运用了多种机器学习分类算法，包括决策树、随机森林、逻辑回归、线性判别分析和人工神经网络等，来对开发者的流失风险进行预测和分类。这些算法的结合使用提高了模型的准确性和鲁棒性。

·模型评估：为了确保模型的预测性能，我们采用了交叉验证、准确率、精确率、召回率和 F1 分数等多种评估指标来验证模型的有效性。

·特征选择：通过使用特征选择技术，如递归特征消除（RFE）和基于模型的特征选择，我们进一步优化了模型的特征集，以提高模型的泛化能力和减少过拟合的风险。

·数据可视化：利用 Matplotlib 和 Seaborn 等数据可视化工具，我们将复杂的数据模式和模型结果转化为直观的图表，便于理解和沟通我们的发现。

通过这些方法的综合应用，我们能够构建出一个能够准确预测开发者流失风险的模型，为开源社区的维护者提供了有力的决策支持工具。

4 健康度量、寿命预测的方法及其效果

4.1 健康度量方法

在预测开发者流失的场景中可以通过以下健康度量方法提取特征：

• 活跃度指标：包括活跃贡献者数量、PR 创建活动、代码质量相关的活动等，这些指标可以反映项目的活跃程度和开发者的参与度。活跃度相关指标如`active_Cl_pr_create_contributor_activity`等。

• 统计特征：对时间序列数据进行特征提取，包括计算各种统计指标，如均值、中位数、标准差、峰度、偏度等。此外，还可以使用信号处理技术来提取频域特征，如傅里叶变换、小波变换等。

• 机器学习模型：使用 XGBoost、RandomForest、AdaBoost、SVM、KNN、Logistic Regression 等机器学习算法对开发者的流失风险进行分类预测。这些模型可以帮助我们从数据中学习并预测流失风险。

• 综合评估：通过组合多个模型的预测结果，可以提高整体性能，并降低过拟合的风险，减少单一分类器预测错误的数量。例如，使用 XGBoost、RandomForest、AdaBoost 三种分类器的综合预测准确率能达到 90%。

4.2 寿命预测方法

在开发者流失预测中，寿命预测方法是指使用统计或机器学习技术来预测开发者在项目中的活跃寿命或留存时间。这些方法可以帮助开源社区管理者了解开发者的潜在流失时间点，从而采取相应的措施来挽留他们。以下是一些可应用于开发者流失预测的寿命预测方法：

• 随机森林（Random Forest）：随机森林是一种集成学习方法，它通过构建多个决策树并输出平均结果来提高预测的准确性。这种方法在处理具有复杂关系的数据集时非常有效，可以用于预测开发者的流失时间。例如，通过分析开发者的行为模式和贡献历史，随机森林模型可以识别出可能导致流失的关键因素，并预测流失的时机。

• 支持向量机（Support Vector Machine, SVM）：SVM 是一种强大的分类技术，它可以在高维空间中找到最优的决策边界。在开发者流失预测中，SVM 可以用来区分活跃开发者和潜在的流失开发者。通过优化 SVM 的参数，如核函数和惩罚参数，可以提高模型的预测性能。

• XGBoost：XGBoost 是一种基于梯度提升的集成学习算法，它在处理大规模数据集时非常高效。XGBoost 通过优化决策树的组合来提高预测的准确性，并且可以通过调整参数来控制模型的复杂度，从而避免过拟合。在开发者流失预测中，XGBoost 可以用来识别影响开发者留存的关键特征，并预测他们的流失时间。

• 麻雀搜索算法（Sparrow Search Algorithm, SSA）：SSA 是一种基于群体智能的优化算法，它模仿麻雀的觅食行为。在开发者流失预测中，SSA 可以用来优化机器学习模型的参数，如 SVM 的核函数和惩罚参数，以提高预测的准确性。通过迭代搜索最优参数，SSA 可以帮助构建更准确的流失预测模型。

• Transformer 模型：在开发者流失预测中，Transformer 可以用于分析开发者的交互模式和行为序列，以预测他们的流失风险。结合 Adaboost 等集成学习方法，可以进一步提高预测的鲁棒性和准确性。

这些方法各有优势，通过综合运用这些技术，可以更有效地预测开发者的流失，从而为开源社区的维护和管理提供支持。

4.3 效果评估

开发者流失预测模型有以下评估指标和方法：

• 准确率（Accuracy）：这是最直观的评估指标，表示模型预测正确的样本数占总样本数的比例。准确率越高，说明模型的预测结果越可靠。

• 精确率（Precision）和召回率（Recall）：精确率是指模型预测为正类别中实际为正类别的比例，而召回率是指所有实际为正类别中被模型正确预测为正类别的比例。这两个指标有助于评估模型在不同类别上的预测性能，特别是在处理不平衡数据集时。

• F1 分数（F1 Score）：F1 分数是精确率和召回率的调和平均数，它综合考虑了精确率和召回率的平衡，是一个在精确率和召回率之间取得平衡的指标。

• ROC 曲线和 AUC 值：ROC 曲线是一个性能度量图表，用于展示模型在不同阈值下的真正例率（召回率）和假正例率（1-特异性）。AUC 值（Area Under the Curve）表示 ROC 曲线下的面积，它衡量模型的整体性能，AUC 值越高，模型的区分能力越强。

• 混淆矩阵（Confusion Matrix）：混淆矩阵用于描述模型预测结果和实际结果之间的关系。它可以帮助我们了解模型在不同类别上的预测准确性，以及模型可能存在的偏差。

• 特征选择的评估标准：在进行特征选择时，我们可以使用 IV 值、相关性分析、PSI 计算、方差分析等方法来评估特征的重要性，并选择对模型性能有显著影响的特征。

• 模型比较：可以通过比较不同模型的性能指标来选择最佳的模型。例如，可以比较随机森林、XGBoost、逻辑回归等不同模型的准确率、F1 分数和 AUC 值，以确定哪个模型最适合当前的任务。

• 时间序列分析：在预测开发者流失时，可以使用时间序列分析方法，如 Prophet 算法，来预测开发者的活跃度和流失趋势。

通过综合使用这些评估指标和方法，可以全面地评估开发者流失预测模型的效果，并根据评估结果对模型进行优化和改进。

5 实验结果

决策树(Decision Tree):

• 准确率:0.76

• 精确率:0.23

• 召回率:0.18

• F1 分数:0.20

决策树的总体准确率不高,但在流失开发者的识别上比逻辑回归稍好.

随机森林(Random Forest):

• 准确率:0.81

• 精确率:0.27

• 召回率:0.14

• F1 分数:0.19

随机森林的总体准确率相比决策树较高，但召回率较差

逻辑回归(Logistic Regression):

• 准确率:0.84

• 精确率:0.71

• 召回率:0.06

• F1 分数:0.11

逻辑回归在总体准确率上表现较好，但召回率和 F1 分数很低，说明此模型识别流失开发者的能力有限和可能存在不平衡数据集问题

6 科学与实际意义

6.1 科学意义

• 跨学科研究：该模型结合了计算机科学、数据科学和社会科学等领域的知识，为跨学科研究提供了一个实证研究平台。

• 算法创新：通过应用和比较不同的机器学习算法（如随机森林、逻辑回归、决策树等），推动了算法的发展和优化。

• 特征工程的重要性：该研究强调了特征工程在机器学习模型中的核心作用，为特征选择和工程提供了新的见解和方法。

6.2 实际意义

• 项目管理：帮助开源项目管理者识别和预防开发者流失，从而更有效地管理项目资源和人力资源。

• 降低风险：通过预测潜在的流失风险，项目团队可以提前采取措施，降低因开发者流失带来的项目风险。

• 资源优化：模型可以帮助项目团队优化资源分配，例如，将资源集中在不大可能流失的关键贡献者身上，以提高资源使用的效率。

• 社区建设：对于开源社区而言，该模型有助于理解社区成员的动态变化，从而制定策略以增强社区的凝聚力和活力。

• 持续改进：模型的预测结果可以作为反馈，帮助项目团队持续改进项目管理实践，提升项目的整体健康度。

• 策略制定：为开源项目的长期战略规划提供数据支持，帮助项目领导者制定更有效的招募和留存策略。

综上所述，开发者流失预测模型不仅在科学上具有探索价值，而且在实际应用中具有深刻意义，对于推动开源社区的可持续发展具有重要作用。